

Inheritance

Inheritance

Inheritance allows one class to be a specialization of another class.

Advantages:

Code reuse: different classes can share code

Abstraction: different classes can be treated the same

Inheritance in C++

```
class Derived : public Base
{
//inherited member variables // from Base
additional member variables;
new constructors and destructor;
//inherited functions; // from Base
overriden functions; // replacing Base's
new functions;
};
```

Access Control

The **Derived** class has direct access to the **public** members of the **Base**, and not to the **private** members.

One can use **protected** in the **Base** to give access to a **Derived** class but not the general user.

Polymorphism

The **Derived** class is a **special** case of the **Base** class. Thus it can be used wherever the **Base** class can. But not vice versa.

```
Base *R = new Derived();    // okay
```

```
Derived *S = new Base();    // not allowed
```

Constructor

Note that a constructor for the **Base** class is ***always*** executed ***before*** the constructor for the **Derived** class.

The default constructor for the **Base** class is used, unless your code specifies otherwise.

Destructors

The destructor for the **Base** class is ***automatically*** executed ***after*** the destructor for the **De-
rived** class.